Emergence and Control of Macro Spatial Structures in Perturbed Cellular Automata, and Implications for Pervasive Computing Systems

Marco Mamei, Andrea Roli, Franco Zambonelli

Abstract—**Predicting the behavior of complex decentralized pervasive computing systems before their deployment in a dynamic environment, as well as being able to influence and control their behavior in a decentralized way, will be of fundamental importance in the near future. In this context, this paper describes the general behavior observed in a large set of asynchronous cellular automata when external perturbations influence the internal activities of cellular automata cells. In particular, we observed that stable macro-level spatial structures emerge from local interactions among cells, a behavior that does not emerge when cellular automata are not perturbed. Similar sorts of macro-level behaviors are likely to emerge in the context of pervasive computing systems and need to be studied, controlled, and possibly fruitfully exploited. On this basis, the paper also reports the results of a set of experiments showing how it is possible to control, in a decentralized way, the behavior of perturbed cellular automata, to make any desired patterns emerge.**

Index Terms **— Cellular Automata, Self-organization, Pervasive Computing, Multiagent Systems.**

I. INTRODUCTION

COMPUTING will soon become pervasive and autonomous. On the one hand, our everyday environments will be more and more populated by multitudes of decentralized and networked computing systems (e.g., multi-agent systems [38], ad-hoc networks of mobile computer-based devices [10], sensor networks [12], clouds of "smart dust" [29], spray computers [39]). On the other hand, most of these systems will be able to autonomously perform activities on our behalf – typically by interacting with each other – letting us "out of the loop" [19].

The potential applications of future pervasive computing scenarios are endless, promising to impact in all our activities. For this reason, it is of dramatic importance to understand and predict how such systems will possibly behave. It has been recently discovered that the Internet, the Web, and the Gnutella network – the only deployed examples of large-scale decentralized and autonomous computing systems – have structurally evolved in rather peculiar and unexpected ways, strongly impacting on reachability of information and reliability [2, 3, 30]. Researchers in the area of pervasive computing should learn from this, and should start asking *now* whether similar situations may occur *in the future* for pervasive computing systems and which tools and methodologies they can exploit to control them.

This paper is a small step in that direction. In particular, we present and discuss a number of experiments that we have performed on a large set of cellular automata (from now on CA) [6, 36], in which we have tried to mimic in a minimalist way the key characteristics likely to be exhibited by future pervasive computing systems:

– *Locality in interactions:* Future pervasive computing systems will typically interoperate based on local patterns, as deriving from, e.g., short-range wireless communications and/or local directed networks. Such characteristic is naturally reflected in CA, which evolve via local interactions in a lattice.
– *Autonomy of components*: Pervasive computing systems tend to be fully decentralized, with components executing in total autonomy. Such characteristics can be reflected by asynchronous CA, i.e., CA in which state transitions in cells occur asynchronously, according to local internal dynamics [17].
– *Perturbation of the environment*: pervasive computing systems will be embedded and interact with dynamic environments that can continuously influence the local activities of distributed components. Such characteristics can be emulated by perturbing the internal state of CA cells.

In a first set of experiments, we show that asynchronous CA, when perturbed, exhibit peculiar interesting behaviors. In particular, during their dynamic evolution, and despite the out-of-equilibrium situation induced by induced by environmental factors outside the CA, stable macro-level spatial patterns may emerge from local interactions among cells, a behavior that does not emerge when the cellular automaton is synchronous

Manuscript received November 20, 2004.

This work is supported by the Italian MIUR and CNR in the "Progetto Strategico IS-MANET, Infrastructures for Mobile ad-hoc Networks".

Marco Mamei is with University of Modena and Reggio Emilia, Modena, Italy (e-mail: mamei.marco@unimo.it).

Andrea Roli is with University "G.D'Annunzio", Pescara, Italy (e-mail: a.roli@unich.it)

Franco Zambonelli is with University of Modena and Reggio Emilia, Modena, Italy (e-mail: zambonelli.franco@unimo.it)

and not perturbed. On this basis, the paper argues that macro-level patterns, similar to the ones exhibited by CA, will be observed, in terms of globally coordinated patterns of activities, as soon as multitudes of interacting computer-based systems will start populating our networks and our physical spaces. These will be likely to dramatically influence the overall behavior of such systems at a very large scale.

A second set of experiments explores how and to which extent it is possible to exert some sort of decentralized control on the perturbed CA, so as to influence its evolution and make a desired pattern emerge. Of course, for such a methodology to be potentially applicable to pervasive and decentralized computing scenarios, it must account for the impossibility to fully control the behavior of each of the components, but only portions of the systems. Two methodologies of these kinds have been experienced, one of which rather successfully, envisioning the possibility of actively controlling the emergence of macro-spatial patterns either defensively, to prevent their possible damaging effects on the system, or constructively, as a tool to enforce global coordination patterns of activities.

The remainder of this paper is organized as follows. Section 2 introduces cellular automata, and characterizes asynchronous dynamics and perturbations. Section 3 presents and analyzes the behaviors observed in perturbed cellular automata. Section 4 discusses the implications of these emergent behaviors for future pervasive computing systems. Section 5 presents the two methodologies we have experienced to control the behaviors emerging from a perturbed CA, and discusses how they may be useful to control the behavior of decentralized pervasive computing systems. Section 6 discusses related works in the area. Section 7 concludes and outlines future work.

II. PERTURBED CELLULAR AUTOMATA

CA are widely used in engineering, physics, biology and social sciences as a model for simulating and studying spatially distributed systems [6]. Indeed, CA represent the essence of distributed systems composed of autonomous entities characterized by local interactions. Furthermore, several researches on emergent phenomena and self-organization very often relies on the computational paradigm of CA, as they provide a simple environment for simulating complex dynamics [36]. A representative example of the application of CA in this perspective is pattern formation, discussed in [6].

In this section, we will shortly introduce the basic CA background, then the specific class of asynchronous CA, and finally the way to induce perturbations on CA.

A. Basic Characteristics

Generally speaking, CA are regular lattices of cells, each one being a finite-state automaton. Starting from an initial global state (determined by all the local states of its cells) a CA dynamically evolves by having cells update their local state depending on a state transition function of their state and of the state of neighboring cells.

More formally, a CA is statically defined by a quadruple $A=(S,d,N,f)$. S is the finite set of possible states a cell can assume. d is the dimension of the automaton, whose cells typically are organized into d-dimensional discrete grid (possibly closed to a d-dimensional torus). N is the neighborhood structure, defining which cells can "influence" each cell in local state transitions. N is typically uniform and isotropic. f is the local transition rule, i.e., a function $f{:}S^N \to S$ mapping a configuration of states in a neighborhood into a state. f is typically the same for each cell (uniform CA).

In this paper, we mostly focus on CA with binary cells ($S=\{0,1\}$, two states that in our figures correspond to a cell being white or black, respectively), arranged in 2-dimensional square grids with wraparound borders ($d=2$). In the following, N will simply indicate the number of neighbors of a cell. With this regard, we adopted the neighborhood structure offering the best compactness to the neighborhood itself (e.g. Moore neighborhood if N is 8, Von Neumann neighborhood if N is 12). In any case, as we will shortly discuss later on, our results apply to a larger class of CA (i.e., to CA with different dimensions, larger state sets for cells, as well as to CA in which cells are connected according to irregular lattices).

Given the quadruple A, that specifies the "static" characteristics of an automaton, we also have to define the update dynamics, in order to have the complete description of a CA. The usual definition of CA (and the most widely studied one) is with synchronous dynamics: cells update their state in parallel at each time step. However, despite being the most studied class, synchronous CA are hardly representative of those real-world phenomena whose execution evolve via the interactions of a population of autonomous interacting elements. In these cases, asynchronous dynamics have to be introduced.

B. Asynchronous Dynamics

Accordingly to the most accepted terminology, a CA is asynchronous if cells can update their state independently from each other, according to a dynamics that can be either step-driven or time-driven [17]. In step-driven dynamics, a kind of global daemon is introduced, whose job is to choose at each time step one (and only one) cell to update, accordingly to a specific sequence. In time-driven dynamics, instead, each cell is assumed to have an "internal clock" which wakes up the cell and makes it update. This is also the case of more interest to us, since in decentralized pervasive computing systems, autonomous processes execute and interact asynchronously accordingly to local internal clocks. Also, time-driven dynamics provides for a more continuous notion of time. In the experiments presented in this paper, CA have an asynchronous time-driven dynamics: at each time step, a cell has a probability λ_a to wake up and update its state. The update of a cell has been implemented as atomic and mutually exclusive among neighbors, without preventing non-neighbor cells to update their state concurrently.

In any case, we emphasize that the choice of the dynamic is

substantial. In fact, asynchronous CA exhibit dynamic behaviors that are very different from the ones of their synchronous counterparts, both in terms of the transient and of the final attractor. Although both synchronous and asynchronous dynamics have the same fixed points [31], the basins of attraction can be very different: some of the final attractors reached under asynchronous dynamics are hardly reached under synchronous one. As an example, Figure 1 compares the global states of two different CA under synchronous and asynchronous dynamics. Beside the visual differences in the perceived patterns, the synchronous regime makes both CA reach a cyclic attractor, while the asynchronous regime make both CA reach fixed-point attractors that their synchronous counterparts have never been observed to be able to reach.

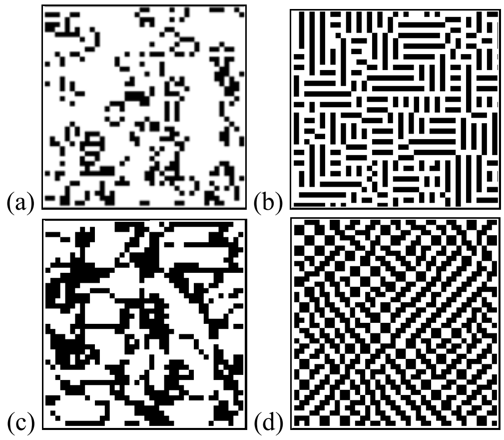


Figure 1: Synchronous vs. asynchronous dynamics in CA. (a) Synchronous CA in one state of a cyclic attractor: S={0-white,1-black}; N = 8; f = {a cell in state S=0 move to state S=1 iff it has 2 neighbors in state S=1; a cell in state S=1 stays in that state iff it has 1 or 2 neighbors in state S=1}. (b) Its asynchronous counterpart in a fixed-point attractor. (c) Synchronous CA in one state of a cyclic attractor: S={0,1}; N = 12; f = {a cell in state S=0 stays move to state S=1 iff it has 6 cells in state S=1; a cell in state S=1 stays in that state iff it has 3,4,5 or 6 neighbors in state S=1}. (d) Its asynchronous counterpart in a fixed-point attractor.

C. *Perturbing CA Evolution*

We are interested in studying the evolution of CA when they are situated, that is, when the internal dynamics of the CA can be influenced by the unpredictable effects of an external environment in which they are assumed to be immersed.

Such situatedness of the CA implies that some cells can be forced by environmental factors outside the CA to change their state (see Figure 2). In a computational perspective, which is that of more relevance here, one must consider that components of pervasive computing systems (agents, embedded sensors, smart dusts) will be typically devoted to monitor and control our physical environments, and will be influenced in their execution by what they sense in such environments. In several cases, environments possess a dynamics that is not controllable or foreseeable. For instance, the temperature and lightening condition in a room that a sensor is devoted to control may vary dynamically for a number of reasons that cannot be predicted.

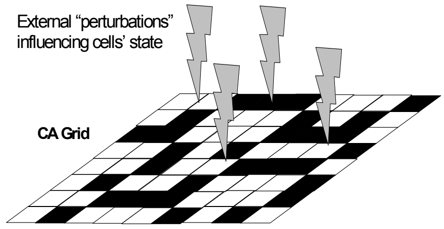


Figure 2: The basic structure of a perturbed cellular automaton.

All the above types of situatedness can be effectively modeled in cellular automata by having external perturbations change the internal states of the CA cells accordingly to specific dynamics. From a more formal point of view, we characterize a perturbed CA as:

- $A = (S, d, N, f)$;
- *asynchronous* time-driven dynamics (with probability λ_a);
- a *perturbation* action $\varphi(\alpha,D)$.

where A is the quadruple defining a CA, the dynamics is the one already discussed in Subsection II.B, and the perturbation action φ is a transition function which acts concurrently with f and can change the state of any of the CA cells to a given state α with some probabilistic distribution D, independently of the current state of the cells and of their neighbors. Specifically, in our experiments with $S=\{0,1\}$, $\alpha=1$ and D is a uniform distribution of rate λ_e.

In our simulations, in which each cell of the asynchronous CA is associated to a separated thread of control, we have implemented the effect of the perturbation dynamics via additional threads of control (asynchronously triggered with probability λ_e) capable of forcing the cells of the CA to change their state, independently of the state transition function and of the internal cells' dynamics (Figure 2).

III. Emergent Behaviors in Perturbed Cellular Automata

The behavior exhibited by perturbed CA is dramatically different from both their synchronous and not-perturbed asynchronous counterparts. To study such behaviors, we conducted a number of experiments consisting in both visually observing the evolution of CA, and in measuring the degree of regularity of the final (stable) configuration (see below).

In particular, our experiments have focused on understanding the impact of the perturbation dynamics (determined by λ_e) against the internal dynamics of asynchronous CA (determined by the rate of cell updates λ_a).

A. *The Effects of the Perturbation Dynamics*

When the external perturbation is high enough to effectively perturb the internal dynamic of the CA, but it is still not prevailing over it so as to make the behavior of the CA almost random (which happens when λ_e is comparable λ_a), peculiar patterns emerge. In particular, we have observed that

the perturbation on the cells induced by environmental factors outside the CA makes the global state of the CA self-organize into large-scale regular spatial structures. Such structures exhibit long-range correlations between the local states of the CA cells, emerged despite the strictly local and asynchronous transition rules. Figure 3 reports several examples of such phenomenon for different types of binary-state CA. There, for each type of CA, we have reported a sample of a typical pattern emerged under a specific value of the ratio λ_e/λ_a. This enables to clearly outline the influence of the ratio λ_e/λ_a and the generality of the observed behaviors, which apply to a wide range of different CA. For all types of CA, the strictly local patterns that emerge in the absence of perturbation (or in the presence of perturbations with a very low dynamic), tend to extend to a larger scale in the presence of (moderated) perturbations. In particular, in Figure 3, one can observe that, depending on the specific type of CA, these structures may sometime extend to the whole CA grid so as to make a unique global pattern emerge (e.g. Figure 3a, 3c, 3d), other times different large-scale structure, not extending to the whole grid, can coexists within it (e.g., Figure 3b, 3e).

The observed large-scale patterns are rather stable despite the continuous effects of the perturbations. However, in some cases, the emergent patterns are part of a dynamic structure that dynamically evolves due to the continuous perturbing effects, while always preserving the overall structure. For instance, the long diagonal stripes in Figure 3d: *(i)* continuously change their micro-level shape, while maintaining the same global structure; and *(ii)* they tend to continuously and slowly translate horizontally in the CA lattice. Should the perturbation vanish after the patterns have already emerged, the emerged global patterns tend to suddenly freeze, but do not disappear.

The phenomenon that characterizes the behavior of perturbed CA can be intuitively described as follows. When the degree of perturbation is null or rather small, each cell reaches a local static equilibrium (or, in some of the observed cases, reaches soon a small, localized, cyclic attractor), which reflects in a global uniform equilibrium of the whole system, characterized by a multiplicity of small localized stable structures. When the ratio λ_e/λ_a ratio increases, the system is kept in a substantial out-of-equilibrium situation, resulting in continuous attempts to locally re-establish equilibrium. This typically ends up with cell groups having found new equilibrium states more robust with regard to the perturbation (or compatible with it). These patterns are those that are able to continuously correct irregularities in their structure induced by the perturbation. Such stable local patterns, spread by the state transition rule, start soon dominating and influencing the surrounding, in a sort of positive feedback, until a large-scale (if not even global-scale) stabilized situation emerge. The large-scale stabilization typically results in visually observable large-scale patterns and measurable highly ordered configurations of the cells (Figure 4). When the degree of perturbation is high enough to avoid local stable situations to persist for enough time, no local structure survive for enough time to enable large-scale structure to emerge, and the situation becomes somewhat "turbulent".

It is also worth noting that, in general, perturbation makes the CA initialization rather irrelevant. Although in principle there could be special initial cases preventing the CA self-organizing activities to be actuated, such special initial cases are disrupted by perturbation in a short time.

The interested reader can refer to the web page *http://www.agentroup.unimore.it/DCA/* to repeat these experiments on-line via a multi-threaded simulation environment developed in Java.

B. Generality of the Observed Phenomenon

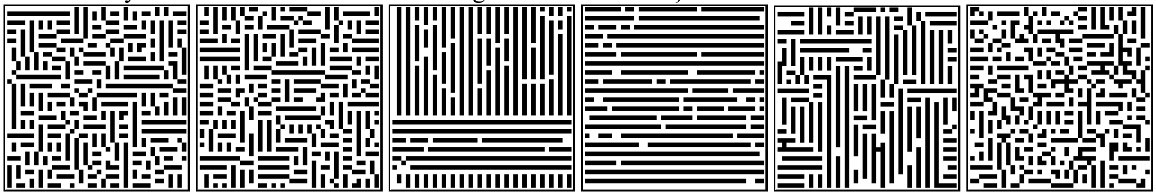
The above behavior is by no mean exhibited by discrete cellular automata with binary state only and connected over a regular lattice.

We have experienced with CA in which the state of cells of a CA can assume values over larger discrete domains or nearly-continuous domains, and have observed that the same qualitative behavior is preserved whenever the state transition function exhibits non-linearity. As an example, Figure 5 shows the results obtained with a CA in which the local state of cells can assume integer values over the range 0—255, and in which a non-linear transition rule is imposed. As the figure outlines, a moderate degree of disturbances promotes the emergence of large-scale "islands" of cells which have somewhat coordinated their behavior.

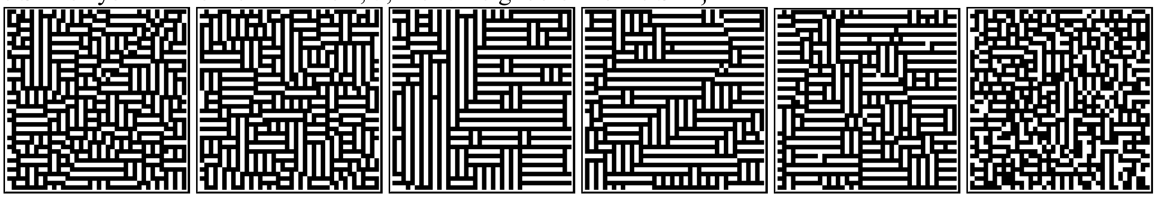
We have also experienced with CA in which cells are not connected in a regular lattice but in an amorphous network, i.e., in which cells are randomly placed in a 2-d space and for which the neighborhood structure is determined by the distances of the cells from each other. Also in this case, as Figure 6 shows, the presence of moderated perturbations makes large-scale patterns emerge that are not observed for a non-perturbed CA. Clearly, because of the amorphous nature of the network (compared to the regular lattice-like structure of the network in "traditional" CA), the patterns that emerge are not geometrically regular.

These results tend to confirm the generality of the phenomenon: the emergence of large-scale patterns from local interactions in the presence of disturbance, independently of the internal structures of cells and of the structure of the network in which they operate. An even more convincing argument of the generality of this phenomenon, however, comes from the observation that very similar phenomena occur in nature, in different classes of physical and biological systems of autonomous components, whenever properly "perturbed" [21, 37].

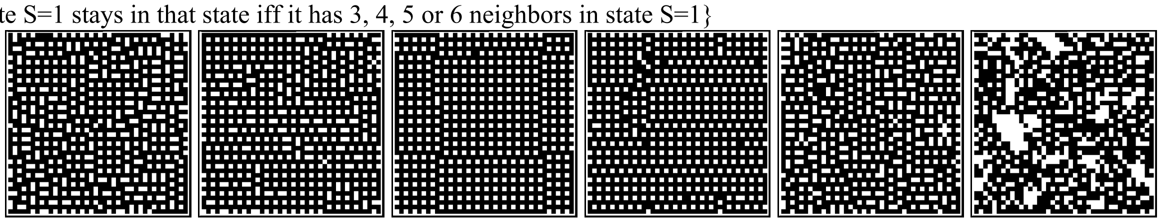
(a) S={0-white,1-black}; N = 8; f = {a cell in state S=0 move to state S=1 iff it has 2 neighbors in state S=1; a cell in state S=1 stays in that state iff it has 1 or 2 neighbors in state S=1}



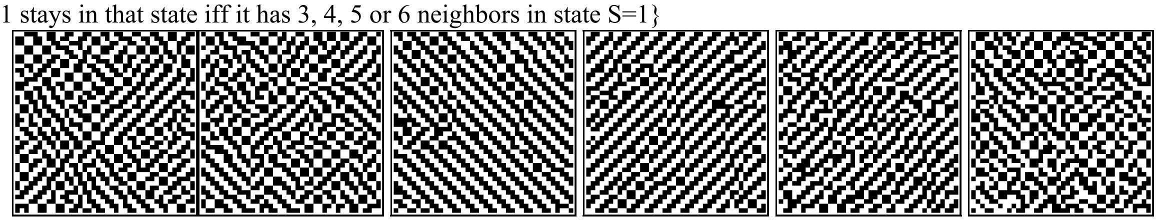
(b) S={0,1}; N = 8; f = {a cell in state S=0 move to state S=1 iff it has 3, 4, or 5 neighbors in state S=1; a cell in state S=1 stays in that state iff it has 2, 3, 4 or 5 neighbors in state S=1}



(c) S={0,1}; N = 8; f = {a cell in state S=0 move to state S=1 iff it has 4, 5, or 6 neighbors in state S=1; a cell in state S=1 stays in that state iff it has 3, 4, 5 or 6 neighbors in state S=1}



(d) S={0,1}; N = 12; f = {a cell in state S=0 move to state S=1 iff it has 6 neighbors in state S=1; a cell in state S=1 stays in that state iff it has 3, 4, 5 or 6 neighbors in state S=1}



(e) S={0,1}; N = 20; f = {a cell in state S=0 move to state S=1 iff it has 6 neighbors in state S=1; a cell in state S=1 stays in that state iff it has 3, 4, 5 or 6 neighbors in state S=1}

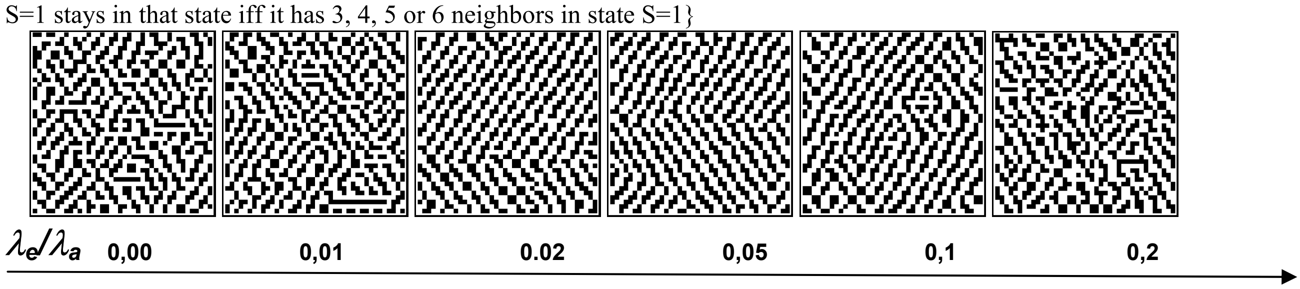


Figure 3: The behavior of different CA depending on the λe/λa ratio.

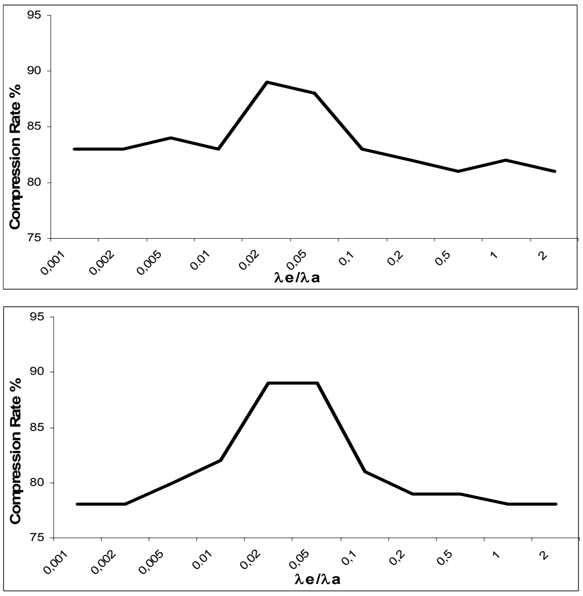
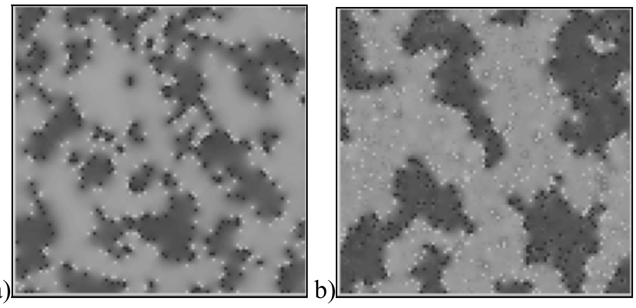
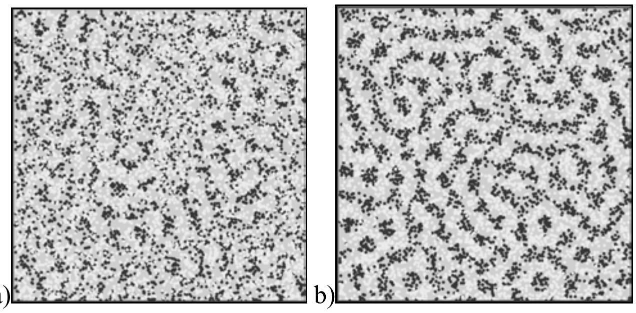


Figure 4: Regularity of the emergent patterns depending on λe/λa ratio, measured by the degree at which the pattern can be compressed. Measures performed with the Lempel-Ziv algorithm (used in WinZip 8.0) on the binary strings representing the rows of the CA. (up): Same type of CA of Figure 3a. (bottom): Same type of CA of Figure 3d.



$\lambda_e/\lambda_a = 0{,}00$ $\lambda_e/\lambda_a = 0{,}02$

Figure 5. Emergence of large-scale patterns in a perturbed nearly-continuous CA with the following characteristics: S={0(white)—255(black)}; N = 8; f = {each cell evaluates its next state NS by comparing its current state CS and the average value A of neighboring cells: if |CS-A|<64 then NS = A, if |CS-A|>64 and A≥128 NS=255, if |CS-A|>64 and A<128 NS=0}. Perturbations set cell state to a random value.



$\lambda_e/\lambda_a = 0{,}00$ $\lambda_e/\lambda_a = 0{,}02$

Figure 6. Emergence of large-scale patterns in an amorphous binary-state CA. Cells are randomly placed in a 2-d space, and the neighborhood of a cell is defined by all those cells that are within a specific distance (in the above test, the distance is set so that a cell is neighbor of 100 cells on average). The transition rule is as follows: f = {a cell in state S=0 stays move to state S=1 iff it has more than 30 and less than 50 neighbors in state S=1; a cell in state S=1 stays in that state iff it has more than 24 and less than 60 neighbors in state S=1}.

I. Implications for Pervasive Computing

As already outlined, perturbed cellular automata have characteristics that will be reflected by future pervasive computing systems: locality in interactions, autonomy of components, and situatedness in a dynamic environment. Despite the minimal model of pervasive computing systems that CA represents, and in consideration of the generality of the observed phenomenon, there are very good reasons to presume that the emergence of global spatial patterns exhibited by CA will be observed – in terms of emergent globally coordinated patterns of activities – as soon as our environments will start being populated by multitudes of dispersed computer-based systems.

These considerations suggest that the design and development of pervasive computing systems (and in general of any other kinds of highly-decentralized software systems) should take into account these phenomena and be ready for:

– A prudential defensive approach, to prevent the emergence of dangerous counterproductive behaviors; or
– A constructive offensive endeavor, to foster the emergence of specific emergence behaviors towards the achievement of otherwise difficult global application goals.

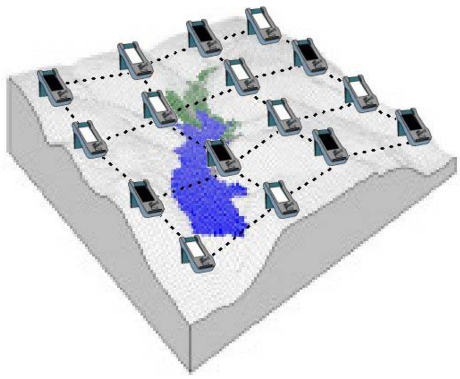


Figure 7. The nodes of the sensor network resemble the cells of a CA.

A. *Defending from Emergent Behaviors*

Defensively, the reported experiments alert on the possibility that a software system immersed in a dynamic environment may exhibit behaviors very different from the ones it was programmed and tested for.

To ground the discussion, let us consider a sensor network application scenario, in which a large number of computer-based devices with sensing capabilities are dispersed in an outdoor environment (see Figure 7). In general, the very goal of a sensor network is to spatially coordinate the activities of the sensors to cooperatively achieve specific sensing activities in an environment. For instance, we can consider that sensors have to coordinate/synchronize their activities so as to: *(i)* perform some collective coordinate sensing to detect events that no single sensor could detect in full by its own; *(ii)* save battery power by sleeping whenever possible, still ensuring that a given geographic zone is always properly monitored by a reasonable number of active sensor.

In this scenario, if unexpected macro-level spatial patterns in the activities of the sensor network emerge, they may cause large amount of data be left out from the network perception. In unlucky situations, a sensor network could miss out a great portion of the events happening in the environment (e.g., if the event is spatially distributed like the emerged macro-level spatial pattern, but in a complementary way).

It is worth noting that these problems are of a very general nature, and similar considerations apply to a variety of other novel pervasive computing scenarios. In the case of information retrieval applications – and specifically of location-based information access – emergent coordination may cause a large amount of available information to be left out of the search, and have the remaining portion over accessed. Emergent coordination in the cells of a mobile telephony system (organized – as a CA – in regular grids) may lead to bandwidth saturation and cause denial of services malfunctioning. In mobile ad-hoc networks, emergent coordination in bandwidth utilization may limit throughput and increase latency in message delivery.

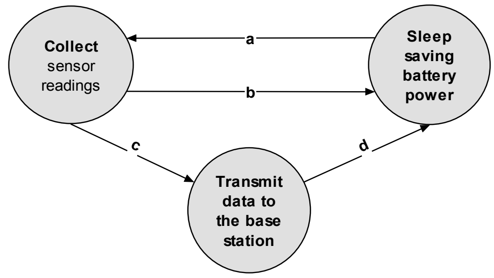
B. Exploiting Emergent Behaviors

The emergence of large-scale or global patterns in a system could also turn, in some situations, to be a useful and desirable characteristic. In the case of sensor networks, for instance, macro-level spatial patterns in the activities of the sensors could enable to easily discover the presence of correlated spatial events happening in the environment.

To test these ideas concretely, we used a network simulator developed within our group [23] to emulate a sensor network scenario: sensors are deployed in an environment in an almost regular formation and can sense the temperature at their location (the temperature values are sampled between 0 and 50 degrees).

On this basis, we developed a simple application enabling the sensor network to detect spatially distributed temperature-changes in the environment, while trying to preserve sensors' battery power. In this application sensors are equally programmed as a finite state automata with three states. In *sleep-state*, the node shuts-down sensors and stops performing high-level tasks to save battery power. In *collect-state*, the node is completely on and perform sensors readings. In *transmit-state* the node sends collected information to a base station for further processing. Each sensor periodically sends its current state to its neighbors and change its state according to the transition rules depicted in figure 8. Transitions are triggered either by the number of neighbor sensors in a specific state and by the value of sensor readings (in figure 8, "boring" temperature readings are below 0.5 degrees, "interesting" readings are above 45 degrees).

Following this model, the state transitions exhibited by the sensor nodes perfectly resembles those of a perturbed cellular automata: nodes communication takes into account the CA internal dynamics, while sensor readings relate to CA external perturbations.



a: if (sleeping neighbors > 3 && 0<collecting neighbors <4)
b: if (collecting neighbors >3 && 0<sleeping neighbors <4) **or if** (boring sensor reading)
c: if (interesting sensor reading) **or if** (neighbors already transmitting > 0)
d: always once complete transmission

Figure 8. The nodes in the sensor network resemble the cells of a CA.

Following the rules in figure 8, macro-level spatial patterns arise in the sensor network. In particular, stripes of coordinated sleeping or collecting activities arise (see figure 9). It is also worth noting that in this realistic implementation the CA cannot be closed in a torus (this in fact would require long-range routing protocols to let nodes at opposite edges to communicate as if they were neighbor). For this reason nodes at the edge of the network have less neighbors than the ones in the middle. Such an asymmetry can create fringe patterns (see the left area of figure 9) .

Moreover, bursts of transmission can be started from a stripe of sensors in the *collecting-state* to send data to the base station. This allows to easily aggregate data and to detect stripe-like distributed events, like a fire spread in a line of trees. With regard to this point, it is particularly interesting to remark that the coupling between nodes' activities and sensors readings (recall that nodes can change state also in response to "boring" or "interesting" sensor readings) also tend to automatically orient the macro-level stripes of nodes' activity to eventual macro-level stripes of physical quantities in the environment. In our opinion, this property shows, in a rather concrete setting, how external perturbation to CA dynamics may be prove useful to realize global coordination patterns.

We are aware that similar types of global coordination of activities in a distributed system can be achieved via specific distributed coordination and synchronization algorithms. For instance, in our example, sensors could achieve strip-like coordination patterns by exploiting self-localization algorithms and subsequently by having sensors move to specific states depending on their local coordinates and on what they sense in the environment. However, achieving a similar behavior by emergence could turn out to be much more efficient and simpler to be programmed. More in general, the possibility of making global patterns emerge in a system from local interactions could be exploited so as to enforce global coordination and synchronization in a large-scale system with very low efforts.

In perspective, while environmental dynamics can plays a major to promote the emergence of large-scale spatial patterns, specific control mechanisms should be envisioned to enable influencing the global behavior of decentralized distributed computing systems from "outside the loop" [12].

That is, to enable promoting/inhibiting the emergence of specific patterns without assuming the capability of intervening directly on each and every running components of system itself.

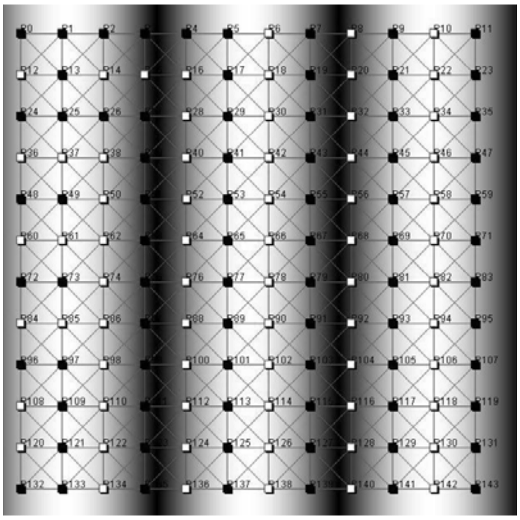


Figure 9. A screenshot of our sensor network scenario. The shade in the background represents the distribution of temperature in the environment (light = hot, dark = cold). Black nodes are in the *collect-state*, white nodes are in the *sleep-state*. In this screenshot, there are not nodes in the *transmit-state*. Macro-level vertical stripes of coordinated activities are forming. The fact that macro-level stripes of nodes' activity and macro-level stripes of temperature values in the environment are aligned, is favored by the coupling between the activity of a node and its temperature measurements.

II. Decentralized Control of CA Behaviors

From the previous discussion, it turns out that it would be of fundamental importance to have the possibility of controlling emergent behaviors in CA. In fact, by considering again the relations with decentralized pervasive computing systems, this may open up the doors to both defending from the emergence of undesirable behaviors and exploiting useful behaviors by making them emerge as needed. However, due to the characteristics of such systems one cannot think at controlling the global behavior of the system via a direct control on all its components. Rather, such control must be as much distributed and decentralized as possible, and should rely only on the possibility of controlling a few components of the systems, without making any assumption on the possibility of controlling all components and their dynamic interactions.

To this end, we experienced two complimentary ways of controlling emergent behaviors in perturbed CA, one of which having lead to quite successful results.

A. *Generalization-based Control*

The generalization-based methodology gets inspiration from Hopfield's work on neural networks [14]. It is recognized that Hopfield's networks can generalize a pattern from imposition of a partial pattern. In CA, super-imposition of a partial pattern translates into initializing a localized sub-set of CA cells according to the desired patterns (see Figure 10 for an example applied to the CA of Figure 3a), and then let the CA evolve by making the global pattern spontaneously emerge from the local imposition.

In the case of decentralized pervasive computing systems, such methodology would imply the possibility of controlling the activities of a local cluster of components, and then let this locally imposed control diffuse to the whole system. This property would be very important in those cases in which only a spatially limited portion of the system is accessible for control and modification. For example, considering the presented sensor network scenario, such an approach would imply controlling some sensors close to an easily accessible base station and have the control diffuse to the whole region.

From our experiments, we found out that, for a few rules and for small grid dimensions, such methodology works quite well (provided that the degree of perturbation dynamics was in the appropriate moderate range). For instance, to make a given pattern emerge out a 20x20 CA grid under the rule of Figures 1b and 3a, one has to initialize a local portion of about 20% of the global grid size. In the case of a pervasive computing system, this would mean that controlling the initial state of a cluster of a few dozens of component may be enough to influence the whole behavior of a system with hundreds of distributed components. Unfortunately, the generalization-based methodology appears to be neither general nor scalable. Generality is lacking because the methodology does not work with all types of CA, but only for those types of CA that exhibit, in the presence of perturbations, globally-extended patterns (e.g., the CA in Figure 3a, 3c, 3d). With regard to scalability, one can see from Figure 11 that, even if with specific rules the methodology can work quite well with small systems, the increase of the system size requires a more than proportional increase in the size of the superimposed pattern. As a further disadvantage, the above methodology cannot be exploited in any way to avoid a specific global configuration to emerge.

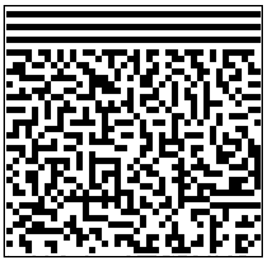


Figure 10. Generalization-based methodology: a local pattern imposed on a portion of a CA grid (same type of CA as in Figures 1a and 3a).

B. *Rule-based Control*

The rule-based methodology starts from totally different considerations and attempts at controlling the global behavior of CA by changing – in a limited percentage of the system cells – the local rules *f* determining their state transitions. With reference to a decentralized pervasive computing system, such control methodology would imply having control over a limited percentage of its distributed components, and would translate in injecting in these components (e.g., via mobile code technologies [10] and wireless communications) some

new working parameters and procedures. Or, in the case of sensor networks, this could imply deploying in the network some additional sensors to complement those already deployed.

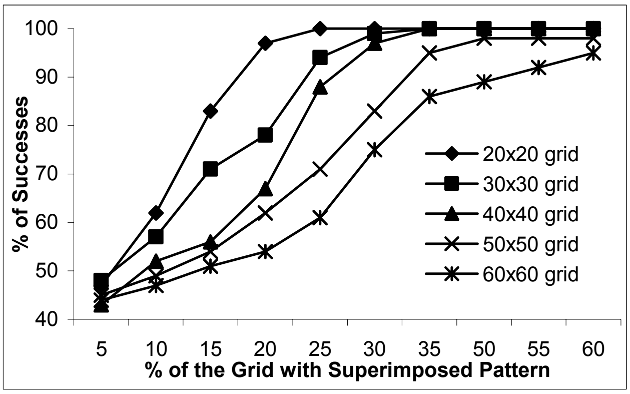


Figure 11. Percentage of successes of the generalization-based methodology, depending on the percentage of superimposed cells, and for different dimensions of the CA grid (same type of CA as in Figures 1a and 3a).

In general, to make a desired pattern emerge in the CA, the rule has to change so as to make (some of the) cells recognize that the cells in the neighborhood are in the right configuration, i.e., their states overall define a local configuration that is compatible with the global one that one is expected to emerge. Among a variety of possible rule modifications tested, we have found any good rule modification should not be too strict, i.e., it should not force a cell to continuously change its state in the attempt at locally forming a suitable configuration. For instance, consider the following modified rule: a cell should stay in its current state if and only if this state, together with the state of neighboring cells, overall defines a local configuration suitable to the emergence of the global one; otherwise it should change its state. For such a strict rule, the whole automaton ends up in a chaotic unstable state (cf. constraint-based CA in [36]). Instead, we have found out that "good" rule modifications have to be very weak and, counter-intuitively, should enable limiting the "normal" state transitions of cells (as implied by the unmodified rule) whenever the state of neighbor cells defines a local configuration that is not in sharp contrast with the desired global one. For instance, given the generic rule for a binary-state CA:

f = *{a cell in state S= 0 must move to state S=1 iff it has between D1 and D2 neighbors in state S=1; a cell in state S=1 remains in that state iff it has between L1 and L2 neighbors in state S=1}*

A simple modified rule *mf* enabling a specific pattern to emerge can be in the form:

mf = *{rule f OR a cell must move to and stay in state S=1 if the neighbor cells currently in state S=1 defines a compatible configuration w.r.t. the desired pattern}*

which can be roughly rephrased as follow: follow the normal transition rule unless the neighborhood situation is already satisfying with regard to the desired pattern.

Applying the *mf* rule (or, in general, a properly modified rule) to a non-perturbed CA, or to a CA perturbed with an inappropriate dynamic, gives no results at all, not even if a very large percentage of cells are modified. Either a local stable configuration is rapidly found by cells (in the case of null or low perturbation dynamics) or the situation becomes, as expected turbulent (in the case of high perturbation dynamics). However, in the presence of moderated perturbation dynamics (in the same range that "normal" not-modified perturbed CA exhibit global or large-scale patterns), the *mf* rule enables any desired pattern to emerge from any initial configuration of any CA, and to extend to a global scale, independently of the chosen *f* rule and independently of the dimension of the CA grid. For instance, we have been able to make the peculiar pattern of Figure 12 emerge from the CA represented in Figure 3d by applying the modified rule, a pattern that emerged only very rarely (about 0,01% of the cases) in previous experiments.

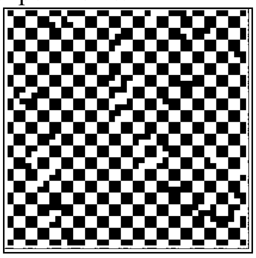


Figure 12. Rule-based methodology: a rare pattern whose emergence can be controlled (same type of CA as in Figures 1c and 3d).

Of course, for a control methodology to be applicable to large distributed and decentralized systems, it must assume the capability of influencing the behavior of only a limited number of the components of a system. For this reason, we have tested the rule-based methodology also by modifying the rule only in a limited percentage of the CA cells. As Figure 13 shows with regard to a specific rule, such experiences have been very satisfying: rule-based control enables a desired pattern to emerge (and extend over the whole grid) even when only a very low percentage of the cells apply the modified rules. In particular, as soon as more than 25% of the cells are controlled, the desired patterns emerge in nearly 100% of the cases. These results are almost independent of the dimension of the grid, making the methodology scalable. Very similar quantitative results apply for different rules.

From the point of view of avoiding a global pattern to emerge, the method is even more effective. In that case, one must apply to a portion of the cells a rule explicitly contrasting the emergence of the undesired pattern, i.e., a rule forcing a cell to change its state whenever the local configuration would be compatible with the global pattern to be avoided:

mf = *{rule f AND a cell must change its state as soon as it recognizes in the neighborhood a local configuration which is compatible to the global one to be avoided}*

By applying such a rule even to a very low percentage of

cells (about 5%) one can avoid emergent patterns to extend to a global scale, although some of them will always be enabled to extend to a rather large-scale. By increasing the percentage of modified cells, emergent patterns are more and more constrained in their possibility to extend to a large scale, until (when the percentage of modified cells grows up to 20%) the effect of perturbations in making large scale patterns emerge is almost annihilated.

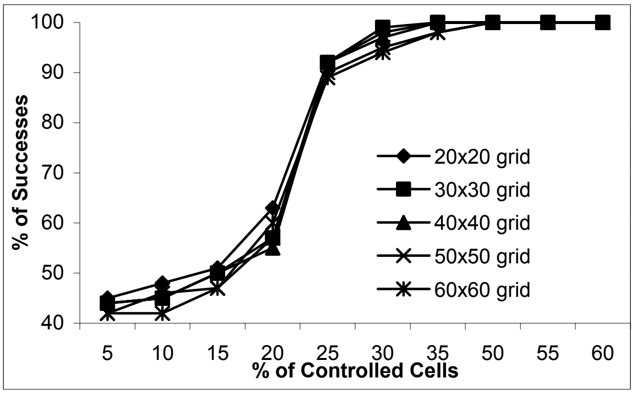


Figure 13. Percentage of Successes of the Rule-based Methodology, depending on the percentage of controlled cells, and for different dimensions of the grid. (same type of CA as in Figures 1c and 3d).

III. Related Works

In this section we briefly discuss related work in the areas of cellular automata, multiagent systems, and pervasive computing systems.

A. Cellular Automata

CA has been extensively studied in the scientific literature, with a specific interest on the very properties of CA, or on their application as computational systems [36], or in the exploitation of CA for simulation purposes [6]. Strictly related, studies in the so called cellular programming area aim at exploiting the emergent behaviors of non-uniform CA in which each cell can have its own local transition function for applications to image recognition, combinatorial optimization problems and evolvable hardware [32]. In the area of simulation [1, 6, 8], CA have been and are still widely exploited for the simulation of biophysical processes and socio-economical phenomena.

In most of the above studies, CA are considered as synchronous and non-situated systems, for the sake of achieving determinism and predictability in CA's behavior (and, thus, in the performed experiments). Some works recognize the peculiar and interesting computational behaviors exhibited by asynchronous model [17, 22, 31, 35]. Nevertheless, they still miss in identifying the strong influences that the situatedness of the system and the perturbation of the environment can have on the behavior of the CA.

Studies on stochastic cellular automata [6] (i.e., CA in which transition rules are probabilistic) have shown that non-determinism in cell update may lead to global scale spatial patterns emerge. Non-determinism can be considered as a sort of situatedness, therefore stochastic automata can be assimilated to perturbed CA in their global dynamics. Nevertheless, studies on stochastic CA lack the explicit identification of non-determinism as a sort of situatedness, and miss in relating them to modern pervasive computing systems. To the best of our knowledge, the only work that identifies the potential relations between stochastic automata and distributed computing systems is described in [7], in which the authors identifies that global scale behavior emerging in stochastic CA can potentially be used as a tool for globally coordinating the behavior of distributed multiagent systems.

B. Multi-agent systems

Since the origins of distributed artificial intelligence and of multiagent systems researches, a large amount of studies have shown that systems in which autonomous components interact with each other in a network, and change their status accordingly to the outcomes of these interactions, can make peculiar global behaviors emerge [13, 15]. Recent examples of these studies may be found in the area of computational markets [20, 34] and of computational ecosystems [16]. However, most of these studies focused on the internal dynamics of the system, without taking into account the perturbation of the environment.

Studies in the area of artificial social laws [24] show that global rules constraining the behavior of all the agents in a group can notably influence the dynamic behavior of the group itself. Analogously, studies adopting an organizational metaphor for the design of multiagent systems [11, 18, 38], show that the definition of global environmental rules to which all agents must obey is very useful toward the effective control of the global multiagent system behavior. The above studies exploit such kind of environmental abstractions constructively during the design process, and assume having full control over the environment behavior. Thus, they miss in identifying that agents may live in dynamic environment, where the rules governing their execution and their interactions can change during the evolution of the multiagent systems and can influence their behavior in unpredictable (or simply uncontrollable) way.

The importance of the environmental abstraction and of its dynamic in the global behavior of the system is properly attributed in the study and implementation of field-based [5] and ant-based multiagent systems [9, 28]. In these systems, very simple agents can indirectly interact with each other in a local way, by spreading synthetic fields or pheromones in the environment and by locally sensing their concentration. Such models can achieve difficult goals, such as: finding shortest paths, clustering data, etc., by exploiting self-organization and emergent phenomena. The similarities between these kinds of multiagent systems and perturbed CA are strong: they both exploit asynchronous components affected in their execution by the environmental dynamics, and both evolve to create global coordinated activity patterns. However, till now, such researches have focused on the possibility of "designing" the environment and of controlling its dynamics to constructively

exploit it. Few researchers explicitly focused on the perturbing effects that uncontrollable environmental dynamics can have on the global behavior of a system [27], and on the possible way to control it.

C. Decentralized Pervasive Computing

Inspired by recent advances in communication and microelectronics technologies, a vast amount of researches propose the exploitation of clouds of small-scale computer-based systems, to be embedded in our everyday objects and to be dispersed in our everyday environments, so as to enrich them with smart sensing and actuating capabilities. The very goal of the applications proposed in this area is somehow related to make the distributed components (nodes, sensors, etc.) coordinate with each others in spatial patterns of activities that are suitable for the achievement of global application-specific tasks.

Researches on amorphous computing [25] aim at devising programming methodologies to enforce specific global behaviors in clouds of amorphously deployed and locally interacting computational particles. There, several algorithms have been proposed to have the particles self-organize into pre-specified spatial patterns of activities.

In the area of sensor networks as well as mobile ad-hoc networking, as already anticipated, one of the mainstream research issues is the enforcement of distributed sensing and activity patterns on the basis of the nodes physical locations. Most of the approaches rely on distributed algorithms to have nodes self-organize a shared coordinate system [4] and rely for all types of spatial activities, e.g., sending messages and data to specific physical regions in the network, spatially differentiate their activities so as save battery power, or sensing in a coordinated way specific spatial patterns of events in the environment.

Strictly related, researches on self-assembly and modular robots [33] are developing similar types of distributed algorithms to have swarms of mobile robots (or of robot components) assume pre-specified spatial shapes.

In any case, in all the above research areas, the main focus is to devise algorithms to obtain specific functionalities by design (i.e. the devised algorithms explicitly and directly addresses the desired functionalities). The possibility that the same functionalities can be obtained, in a simpler and more efficient way by adopting an emergent approach, similar to the one we propose in this paper (i.e. relying on a bottom-up, trial-and-error design), is mostly neglected. Moreover, to the best of our knowledge, none of the above research areas take into consideration the possibility that not-designed, counterproductive and dangerous behaviors, can emerge in the system. On the contrary, the analysis and the experiments reported in this paper alert that this possibility is concrete.

In summary, by considering the strict similarities between the above scenarios and the one of perturbed cellular automata, we expect that analysis of emergent behaviors and tools to control them, possibly along the lines we have sketched in this paper, will soon be required in the domain of decentralized pervasive computing.

IV. Conclusions And Future Work

This paper has reported the outcomes of a set of experiments that we have performed on a large set of asynchronous CA by super-imposing an external perturbation that dynamically influences the state transitions of cells. The experiments have shown that the perturbation makes large-scale spatial structures emerge, a behavior which is not observed under synchronous and unperturbed regime. Starting from that observation, the paper has argued that, since future decentralized and pervasive computing systems will exhibit all of the characteristics of perturbed CA (local interactions, autonomy of components, and influence by a dynamic environment), they will also very likely exhibit similar emergent behaviors. This calls for appropriate models, methodologies and tools, explicitly taking into this emergent phenomenon and possibly exploiting it offensively, as a way to effectively achieve global-scale coordination in a system, or defensively, to prevent and control the emergence of undesired large-scale coordination patterns of activity.

In this context, a further set of experiments have shown how it is possible to control the emergence of large-scale spatial pattern in a perturbed CA by means of a fully decentralized mechanism. We argue that a similar sort of control can be applied to control the behavior of pervasive and decentralized computing systems.

Our results, although applied to a very minimal model of pervasive computing system, motivate further work and experiments. In particular:

- more experiments are needed to evaluate the behavior of CA under a variety of different perturbation regimes, other than the simple ones we have discussed;
- more extensive experiments are needed to evaluate the behavior of more complex CA, i.e., CA with more complex transition rules and in which cells may have differentiated behaviors and may be connected according to a variety of irregular patterns (e.g., amorphous CA and small-world CA);
- strictly related, we have to evaluate the applicability of the proposed control methodologies (discussed in this paper with regard to simple and regular CA) in the context of more complex and amorphous CA.

The main objective is to make our experiments more and more approximate the actual characteristics of real-world pervasive computing scenarios and, eventually, to end up with a powerful simulation environment and with effective general-purpose tools to control the global behavior of large scale decentralized pervasive computing systems.

References

[1] S. Adachi, F. Peper, J. Lee, "Computation by Asynchronously Updating Cellular Automata", Journal of Statistical Physics, 114(1-2):261-289, Jan 2004.

[2] R. Albert, H. Jeong, A. Barabasi, "Diameter of the World Wide Web", Nature, 401:130-131, 9 Sept. 1999.

[3] R. Albert, H. Jeong, A. Barabasi, “Error and Attack Tolerance of Complex Networks”, Nature, 406:378-382, 27 July 2000.

[4] J. Bachrach, R. Nagpal, M. Salib, H. Shrobe, "Experimental Results and Theoretical Analysis of a Self-Organizing Global Coordinate System for Ad Hoc Sensor Networks", Telecommunications Systems Journal, Special Issue on Wireless System Networks, 26(2-4):213-233, Kluwer Academic Publishing, 2003.

[5] S. Bandini, S. Manzoni, G. Vizzari, “Towards a Specification and Execution Environment for Simulations based on MMASS: Managing at-a-distance Interaction”, AT2AI-4, Vienna (A), 2004.

[6] Y. Bar-Yam, “Dynamics of Complex systems”, Addison-Wesley, 1997.

[7] T. D. Barfoot, G. M. T. D’Eleuterio, “Multiagent Coordination by Stochastic Cellular Automata”, Proceeding of the Joint International Conference on Artificial Intelligence, Seattle (WA), Aug. 2001.

[8] S.Bandini, B.Chopard, M.Tomassini. Proceedings of the 5rd Conference on Cellular Automata for Research and Iindustry. Springer-Verlag, 2002.

[9] E. Bonabeau, M. Dorigo, G. Theraulaz. “Swarm Intelligence. From Natural to Artificial Systems”. Oxford University Press, 1999.

[10] C. Borcea, et al., “Cooperative Computing for Distributed Embedded Systems”, 22th International Conference on Distributed Computing Systems, Vienna (A), IEEE CS Press, July 2002.

[11] G. Cabri, L. Leonardi, M. Mamei, F. Zambonelli, “Location-dependent Services for Mobile Users”, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems And Humans, Vol. 33, No. 6, pp. 667-681, November 2003.

[12] D. Estrin, D. Culler, K. Pister, G. Sukjatme, “Connecting the Physical World with Pervasive Networks”, IEEE Pervasive Computing, 1(1):59-69, Jan. 2002.

[13] L. Gasser, J. P. Briot, “Object-based Concurent Programming and Distributed Artificial Intelligence”, in Distributed Artificial Intelligence: Theory and Practice, Kluwer Academic, pp. 81-107, 1992.

[14] J.J. Hopfield, “Neural Networks and Physical Systems with Emergent Collective Computational Abilities", Proceedings of the National Academy of Science USA, 79:2554-2558, 1982.

[15] B. A. Hubermann, T. Hogg, “The Emergence of Computational Ecosystems”, in SFI Studies in the Science of Complexity, Vol. V, Addison-Wesley, 1993.

[16] M. Huhns, "Interaction-Oriented Programming", 1st International Workshop on Agent-Oriented Software Engineering, LNCS No. 1957, Jan. 2001.

[17] T. E. Ingerson, R. L. Buvel, “Structure in Asynchronous Cellular Automata”, Physica D, 10:59-68, 1984.

[18] N. Jennings, S. Bussmann, “Agent-based control systems”, IEEE Control Systems Magazine 23 (3) 61-74, 2003.

[19] J. Kephart, D. M. Chess, "The Vision of Autonomic Computing", IEEE Computer, 36(1):41-50, Jan. 2003.

[20] J. O. Kephart, J. E. Hanson, A. R. Greewald, “Dynamic Pricing by Software Agents”, Computer Networks, 32(6): 731-752, 2000.

[21] Kip, D., Soljacic, M., Segev, M., Eugenieva, E. & Chirstodoulies, D. N. Modulation Instability and Pattern Formation in Spatially Incoherent Light Beams, Science, 290:495-498 (2000).

[22] E. D. Lumer, G. Nicolis, “Synchronous Versus Asynchronous Dynamics in Spatially Distributed Systems”, Physica D, 71:440-452, 1994.

[23] M. Mamei, F. Zambonelli, “Programming Pervasive and Mobile Computing Applications with the TOTA Middleware”, IEEE Conference on Pervasive Computing and Communications (Percom), IEEE CS Press, Orlando (FL), USA, March, 2004.

[24] Y. Moses, M. Tenneholtz, “Artificial Social Systems”, Computers and Artificial Intelligence, 14(3):533-562, 1995.

[25] R. Nagpal, `”Programmable Self-Assembly Using Biologically Inspired Multiagent Control’”, 1st International Conference on Autonomous Agents and Multiagent Systems, ACM Press Bologna (I), 2002.

[26] G. Nicolis, I. Prigogine, Exploring Complexity: an Introduction, W. H. Freeman (NY), 1989.

[27] V. Parunak, S. Bruekner, J. Sauter, "ERIM’s Approach to Fine-Grained Agents", NASA/JPL Workshop on Radical Agent Concepts, Greenbelt (MD), Sept. 2001.

[28] V. Parunak, “Go to the Ant: Engineering Principles from Natural Agent Systems”, Annals of Operations Research, 75:69-101, 1997.

[29] K. Pister. B. Warneke, M. Last. B. Leibowitz, “Smart Dust: Communicating with a Cubic-Millimeter Computer”, IEEE Computer, 34(1):44-51, Jan. 2001.

[30] M. Ripeani, A. Iamnitchi, I. Foster, “Mapping the Gnutella Network”, IEEE Internet Computing, 6(1):50-57, Jan.-Feb. 2002.

[31] B. Schönfisch, A. De Roos, “Synchronous and Asynchronous Updating in Cellular Automata”, BioSystems, 51(3):123-143, 1999.

[32] M. Sipper. “The Emergence of Cellular Computing”. IEEE Computer, 37(7):18-26, July 1999.

[33] K. Stoy, R. Nagpal, “Self-Reconfiguration Using Directed Growth”, 7th International Symposium on Distributed Autonomous Robotic Systems, Toulouse (F), 2004.

[34] L. Tesfatsion, "Agent-Based Computational Economics: Growing Economies from the Bottom Up", Artificial Life, Vol. 8(1), pp. 55-82, MIT Press, 2002.

[35] Y.Wei, S.Ying, Y.Fan, B.Wang, “The Cellular Automaton Model of Investment Behavior in the Stock Market”, Physica A, 325(3-4), 2003.

[36] S. Wolfram, A New Kind of Science, Wolfram Media, Inc., 2002.

[37] J. T. Wootton, “Local Interactions Predict Large-scale Patterns in Empirically Derived Cellular Automata”, Nature, 413: 841:844, 2001.

[38] F. Zambonelli, N. R. Jennings, M. J. Wooldridge, “Developing Multiagent Systems: the Gaia Methodology”, ACM Transactions on Software Engineering and Methodologies, 12(3):417-470, July 2003.

[39] F. Zambonelli, M-P. Gleizes, M. Mamei, R. Tolksdorf, “Spray Computers: Frontiers of Self-Organization for Pervasive Computing”, 14th IEEE Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, IEEE CS Press, Modena (I), June 2004.

Marco Mamei is research associate at the University of Modena and Reggio Emilia, where he received the PhD in computer science in 2004. His current research interests include distributed and pervasive computing, swarm intelligence, and multiagent systems. He is a member of the IEEE, AIIA and TABOO.

Andrea Roli is a research associate at University “D'Annunzio” (Pescara - Italia). He got his Laurea degree in Electronic Engineering at the University of Bologna (Italy) in 1998. He received his PhD in Computer Science from the same university in 2003. His main research subject concerns metaheuristics and complex systems, with applications to optimization problems and multiagent systems.

Franco Zambonelli is professor in Computer Science at the University of Modena and Reggio Emilia since 2001. He obtained the Laurea degree in Electronic Engineering in 1992, and the PhD in Computer Science in 1997, both from the University of Bologna. His current research interests include: distributed and pervasive computing, agent-oriented software engineering, self-organization in distributed systems engineering. In these areas, he has published over 120 papers in international fora, co-edited 7 books, received several best paper awards, and has been invited speaker and tutorialist in several international conferences and workshops. He is a member of IEEE, ACM, AIIA, and TABOO.